

CLAIMS

What is claimed is:

- 1 1. A computer-implemented method for optimization of an executable program that
 2 calls procedures in a shared library, comprising:
 3 identifying linkage code segments in the executable program, wherein each linkage
 4 code segment is associated with a call to a procedure in the shared library, reads procedure
 5 address information from a linkage table, and transfers control to an associated one of the
 6 procedures;
 7 reading the address information from the linkage table;
 8 generating respective substitute code segments for the linkage code segments, each
 9 substitute code segment having references to the linkage table replaced by direct loads of
 10 the address information without reference to the linkage table; and
 11 executing the substitute code segments instead of the linkage code segments.
- 1 2. The method of claim 1, further comprising:
 2 allocating relocation address space for the executable program;
 3 storing the substitute code segments in the relocation address space.
- 1 3. The method of claim 1, further comprising for each calling procedure that calls a
 2 called procedure in the shared library:
 3 identifying registers that are not used by the calling procedure;
 4 generating in a substitute code segment instructions that store in the registers the
 5 address information read from the linkage table.
- 1 4. The method of claim 1, further comprising:
 2 annotating linkage code segments in the executable program during compilation;
 3 placing breakpoints at locations of the linkage code segments as indicated by the
 4 annotations; and
 5 upon encountering a breakpoint at a linkage code segment, generating a substitute
 6 linkage code segment.
- 1 5. The method of claim 4, further comprising:

10004015-101801

10012819-1

identifying in the annotations registers that are not used by a calling procedure that calls a called procedure in the shared library; and
generating in a substitute code segment instructions that store in the registers indicated by the annotations the address information read from the linkage table .

6. The method of claim 1, wherein the executable program includes a calling procedure that calls a first procedure and a second procedure in a first shared library and calls a third procedure in a second shared library, further comprising:
identifying a set of registers that are not used by the calling procedure;
generating in a substitute code segment instructions that store in the registers in the set the address information read from the linkage table
configuring a substitute code segment associated with the first procedure to load the address information from one of the registers in the set into a branch register prior to transferring control to the first procedure; and
eliminating from a substitute code segment associated with the second procedure, an instruction that loads the address information into the one of the registers if the second procedure is called after the first procedure and before the third procedure.

7. The method of claim 1, wherein the executable program includes a call to a first procedure in a first shared library that calls a second procedure in a second shared library, further comprising in the substitute code segment associated with the call by the first procedure to the second procedure, replacing an indirect branch instruction that targets the second procedure with an instruction-pointer relative branch instruction that targets the second procedure.

8. The method of claim 1, further comprising:
placing breakpoints at procedure entry points;
upon encountering a breakpoint at the entry point of a procedure during program execution, searching for linkage code segments in the procedure, generating substitute linkage code segments for those identified in the procedure, and continuing with execution of the procedure.

9. The method of claim 8, further comprising:

10012819-1

- 2 allocating relocation address space for the executable program;
- 3 storing the substitute code segments in the relocation address space.

1 10. The method of claim 8, further comprising for each calling procedure that calls a
2 called procedure in the shared library;

- 3 identifying registers that are not used by the calling procedure;
- 4 generating in a substitute code segment instructions that store in the registers the
- 5 address information read from the linkage table.

1 11. The method of claim 8, wherein the executable program includes a call to a first
2 procedure in a first shared library that calls a second procedure in a second shared library,
3 further comprising in the substitute code segment associated with the call by the first
4 procedure to the second procedure, replacing an indirect branch instruction that targets the
5 second procedure with an instruction-pointer relative branch instruction that targets the
6 second procedure.

1 12. An apparatus for optimization of an executable program that calls procedures in a
2 shared library, comprising:

- 3 means for identifying linkage code segments in the executable program, wherein
- 4 each linkage code segment is associated with a call to a procedure in the shared library,
- 5 reads procedure address information from a linkage table, and transfers control to an
- 6 associated one of the procedures;

7 means for reading the address information from the linkage table;

8 means for generating respective substitute code segments for the linkage code
9 segments, each substitute code segment having references to the linkage table replaced by
10 direct loads of the address information without reference to the linkage table; and

11 means for executing the substitute code segments instead of the linkage code
12 segments.